“NYC TAXI TRIP TIME PREDICTION”

Capstone Project

Submitted To

AlmaBetter

Submitted By:

Vikas panchal
Email id: - panchalvicky501@gmail.com

Naveen Kumar batta
Email id: - naveenbatta4587@gmail.com

Abstract

The objective of our model is to predict the accurate trip duration of a taxi from one of the pickup locations to another drop-off location. In today's fast-paced world, where everyone is short of time and is always in a hurry, everyone wants to know the exact duration to reach his/her destination to carry ahead of their plans. So, for their serenity, we already have million-dollar startups such as Uber and Ola where we can track our trip duration. As a result of this, we proposed a technique in which every cab service provider can give exact trip duration to their customers taking into consideration the factors such as traffic, time, and day of pickup. So, in our methodology, we propose a method to make predictions of trip duration, in which we have used several algorithms, tune the corresponding parameters of the algorithm by analysing each parameter against RMSE and predict the trip duration. To make our prediction we used Random Forest, Decision Trees, and Linear Regression. We improved the accuracy by tuning hyper-parameters and Random Forest gave the best accuracy. We also analysed several data mining techniques to handle missing data, remove redundancy and resolve data conflicts.

Table of Contents

Chapter 1. Introduction:

1.0 Introduction:

There are many possible methods of moving between two given points in a city; however, the taxi trip has found wider applications in urban cities when compared to any other mode of transport. It hence becomes very important to analyze and predict trip duration between two points in the city when provided with the required set of parameters that affect the trip duration. For a good taxi service and its integration with the existing transportation system the project serves as a right means to comprehend the traffic system in the New York City. For prediction purposes factors such as pick up latitude, pick up longitude, drop off latitude, drop off longitude etc. is considered. These geographical locations clubbed with other important factors such as pick up date, pick up time are used for the overall trip duration prediction. The primary focus of this project is in depth analysis of the factors associated with a taxi trip in NYC. The different algorithms used are: Linear Regression, Random Forest, and Decision Trees.

More than 7 billion people exist on earth. With necessities of food, water and shelter there also a key requirement of commutating from one place to other. Rapid advancement in technology in the last two decades leads to adaption of a more efficient way of transportation via internet and app-based transport system. New York city is one of such advanced city with extensive use of transportation via subways, buses and taxi services. New York has more than 10,000 plus taxi and nearly 50% of population doesn't have a personal vehicle. Due to this facts most people used taxi has a there primary mode of transport and it accounts for more than 100 million taxi trips per year.

1.2 Problem statement:

We have the dataset Which is based on the 2016 NYC Yellow Cab trip record data made available in Big Query on the Google Cloud Platform. The data was originally published by the NYC Taxi and Limousine Commission (TLC). The data was sampled and cleaned for the purposes of this project. Based on individual trip attributes, we should predict the duration of each trip in the test set.

Chapter 2

2. Data Summary –

2.1 Import the Dataset –

Before building any machine learning model, it is vital to understand what the data is, and what are we trying to achieve. Data exploration reveals the hidden trends and insights and data pre-processing makes the data ready for use by ML algorithms.

So, let's begin. . .

To proceed with the problem dealing first we will load our dataset that is given to us in a .csv file into a **data frame**.

Mount the drive and load the csv file into a data frame.

```
[ ] #reading dataset
    df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/data/Regression nyc taxi trip time prediction/NYC Taxi Data.csv')
    df.sample(5)
```

	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_fwd_flag	trip_duration
609396	id2971607	2	2016-04-05 12:19:32	2016-04-05 12:53:13	1	-73.963516	40.757416	-73.982880	40.752056	N	2021
862463	id1313813	2	2016-01-30 22:41:59	2016-01-30 22:54:04	5	-73.989578	40.719151	-73.973663	40.743626	N	725
127974	id0946776	1	2016-02-08 08:48:23	2016-02-08 09:23:02	1	-73.983711	40.761040	-73.913086	40.770306	N	2079
136788	id3774564	2	2016-05-16 18:17:57	2016-05-16 18:43:15	2	-73.984589	40.759151	-73.963066	40.775158	N	1518
1316242	id0385001	2	2016-05-01 11:03:30	2016-05-01 11:35:24	2	-73.999123	40.739311	-73.901688	40.693008	N	1914

Fig 2. import dataset.

2.2 Exploratory Data Analysis (EDA) –

We begin our EDA by first checking the distribution of our dependent variable i.e. trip duration. We observed that the data is highly positively skewed. We also plotted the box plot and observed that there are many outliers present in the variable. To cross check this trip duration we have calculated the difference in pick and drop off timing and matched with trip duration we observed no difference. Thus, there is no miscalculation or falsified entries. To eliminate the outliers, we have segregated the data variable into different segment and observed that majority of trip duration is within an hour some observations are within two days but a very few observation are having more than two days. We eliminate such values from out dataset.

We removed id variable as it doesn't give much interpretation. We then calculated the distance based on haversine formula from pickup and drop-off latitude and longitude. Then we plotted the box plot for the variable and observed there are many outlier so we segregate this variable and see that most of the trip are within 10km, some trip are

within 50km while a very few trip crosses 50km. so we eliminate trip with 0 and above 50km distance.

We then checked for categorical variable store_and_fwd_flag and passenger_count. We observed the store and fwd. flag contain majority of one category. So we drop this feature. Passenger count variable has entries from 0 to 9. Since there is no trips with 0 passenger either this a miss entry or the driver forgot to enter passenger count of that trip. Also, in a taxi maximum six person are allowed to sit including minor. So we eliminate 0 and 7-9 records from our dataset.

2.2.1 Dimension of dataset:

```
[ ] #Shape of data

    print ('No. of Examples : ',df.shape[0])
    print ('No. of Features : ', df.shape[1])

    No. of Examples :  1458644
    No. of Features :  11
```

2.2.2 Data-Description:

- id - a unique identifier for each trip
- vendor_id - a code indicating the provider associated with the trip record.
- pickup_datetime - date and time when the meter was engaged.
- dropoff_datetime - date and time when the meter was disengaged.
- passenger_count - the number of passengers in the vehicle (driver entered value)
- pickup_longitude - the longitude where the meter was engaged.
- pickup_latitude - the latitude where the meter was engaged.
- dropoff_longitude - the longitude where the meter was disengaged.
- dropoff_latitude - the latitude where the meter was disengaged.
- store_and_fwd_flag - This flag indicates whether the trip record was held in vehicle memory before sending to the vendor because the vehicle did not have a connection to the server - Y=store and forward; N=not a store and forward trip
- trip_duration - duration of the trip in seconds.

Missing Values:- Find the missing values given dataset.

```
[ ] #checking missing values

    df.isnull().sum()

    id                  0
    vendor_id           0
    pickup_datetime     0
    dropoff_datetime    0
    passenger_count     0
    pickup_longitude    0
    pickup_latitude     0
    dropoff_longitude   0
    dropoff_latitude    0
    store_and_fwd_flag  0
    trip_duration       0
    dtype: int64
```

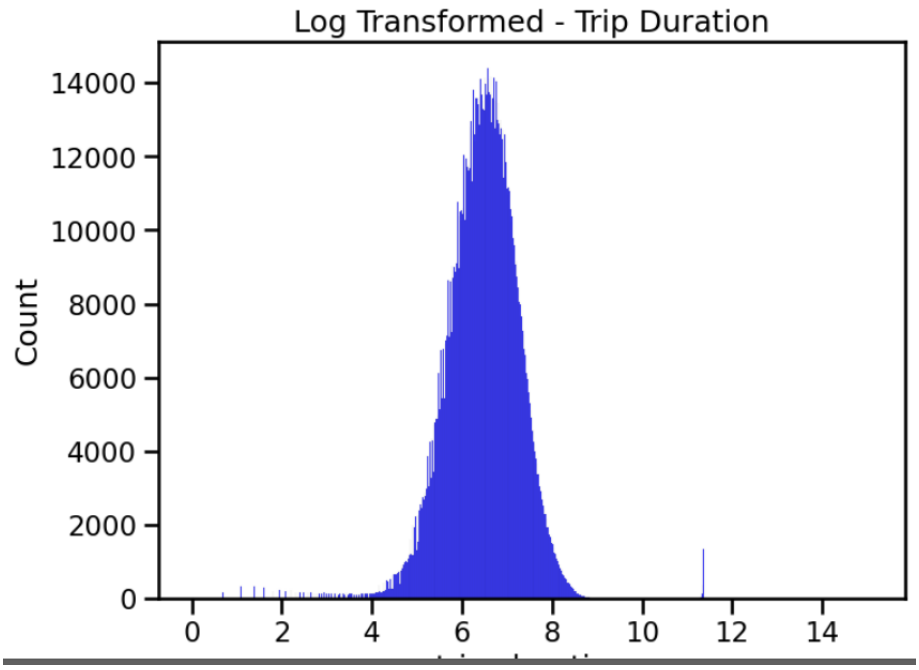
2.2.3 Data Preprocessing :

The info() method prints information about the NYC taxi time dataframe. The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).

```
] #Attribute information

  df.info()

  <class 'pandas.core.frame.DataFrame'>
  RangeIndex: 1458644 entries, 0 to 1458643
  Data columns (total 11 columns):
   #   Column              Non-Null Count    Dtype
  ---  ------              --------------    -----
   0   id                  1458644 non-null  object
   1   vendor_id           1458644 non-null  int64
   2   pickup_datetime     1458644 non-null  object
   3   dropoff_datetime    1458644 non-null  object
   4   passenger_count     1458644 non-null  int64
   5   pickup_longitude    1458644 non-null  float64
   6   pickup_latitude     1458644 non-null  float64
   7   dropoff_longitude   1458644 non-null  float64
   8   dropoff_latitude    1458644 non-null  float64
   9   store_and_fwd_flag  1458644 non-null  object
   10  trip_duration       1458644 non-null  int64
  dtypes: float64(4), int64(3), object(4)
  memory usage: 122.4+ MB
```

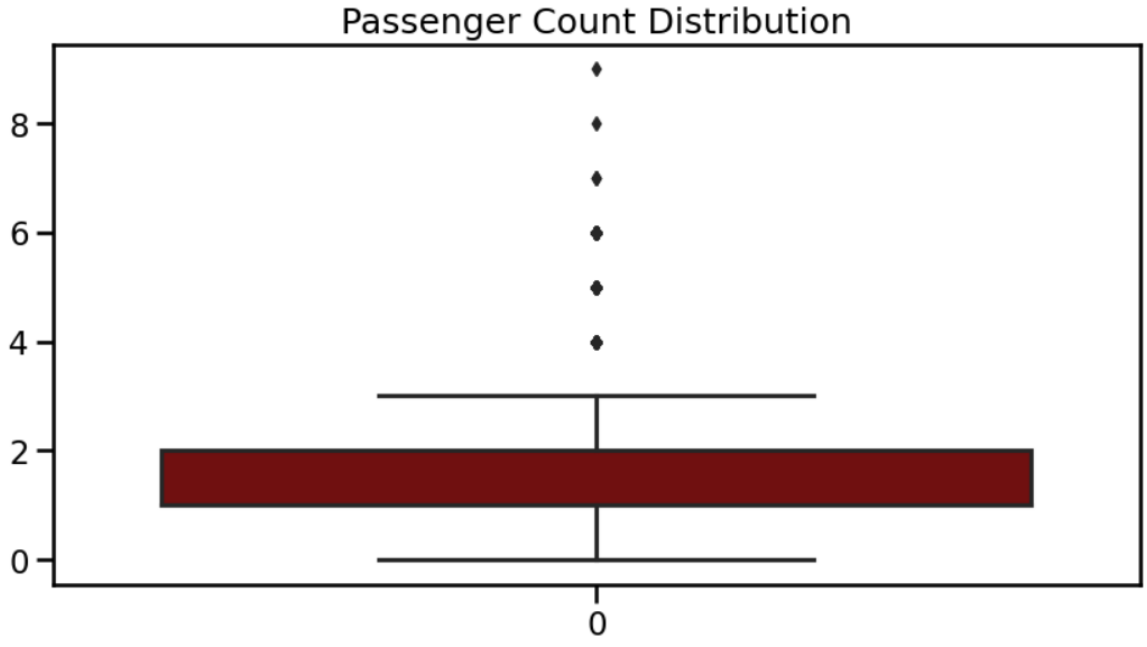
2.2.4 Trip duration data analysis:

- Since our Evaluation Metric is RMSLE, we'll proceed further with Log Transformed "Trip duration".
- Log Transformation Smoothens outliers by proving them less weightage.



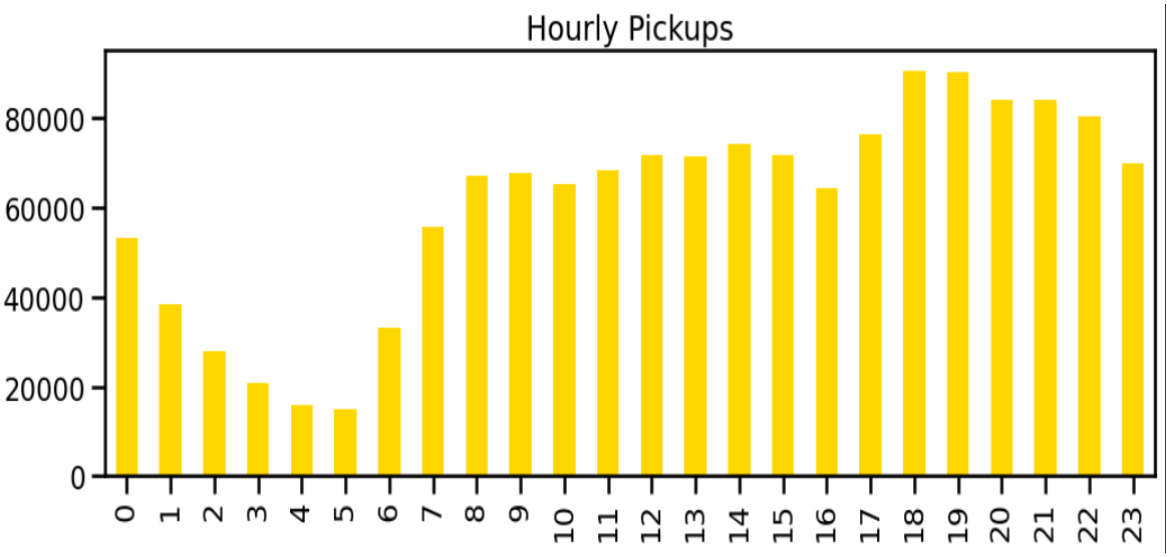
2.2.5 Passenger Count Distribution:

- Most number of trips are done by 1-2 passenger(s).
- But one thing is Interesting to observe, there exist trip with Zero passengers, was that a free ride? Or just a False data recorded?
- Above 4 Passengers Indicate that the cab must be larger sized.

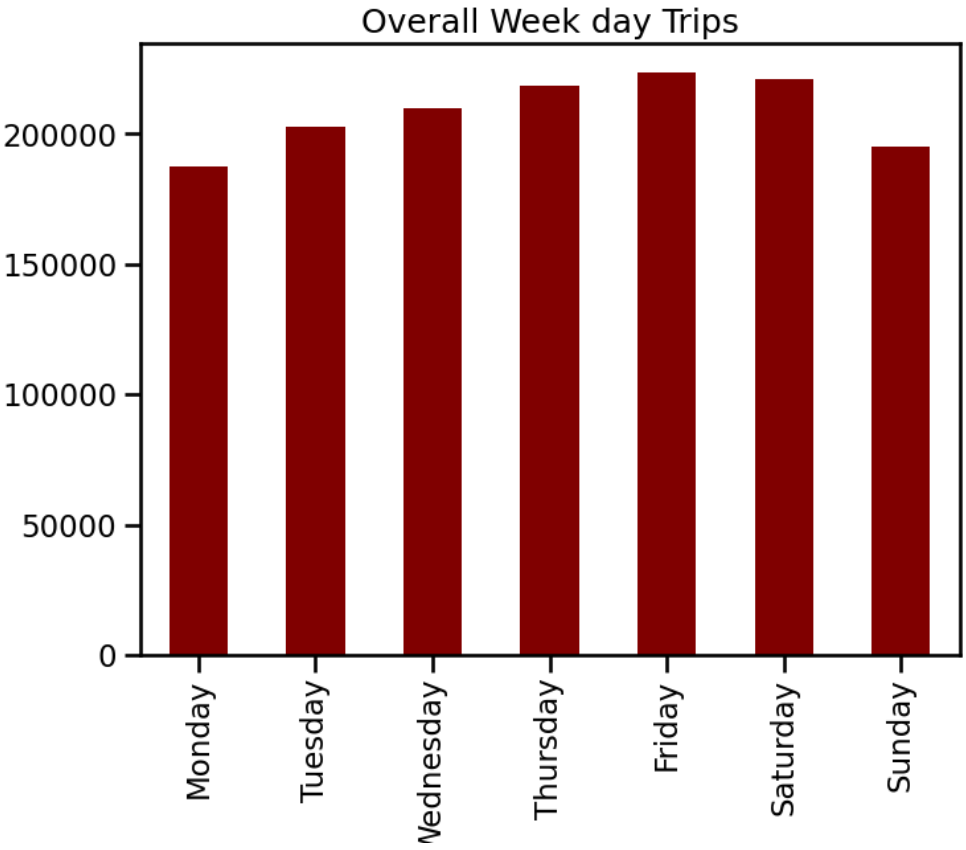


2.2.6 Pickups (Hourly & Weekdays):

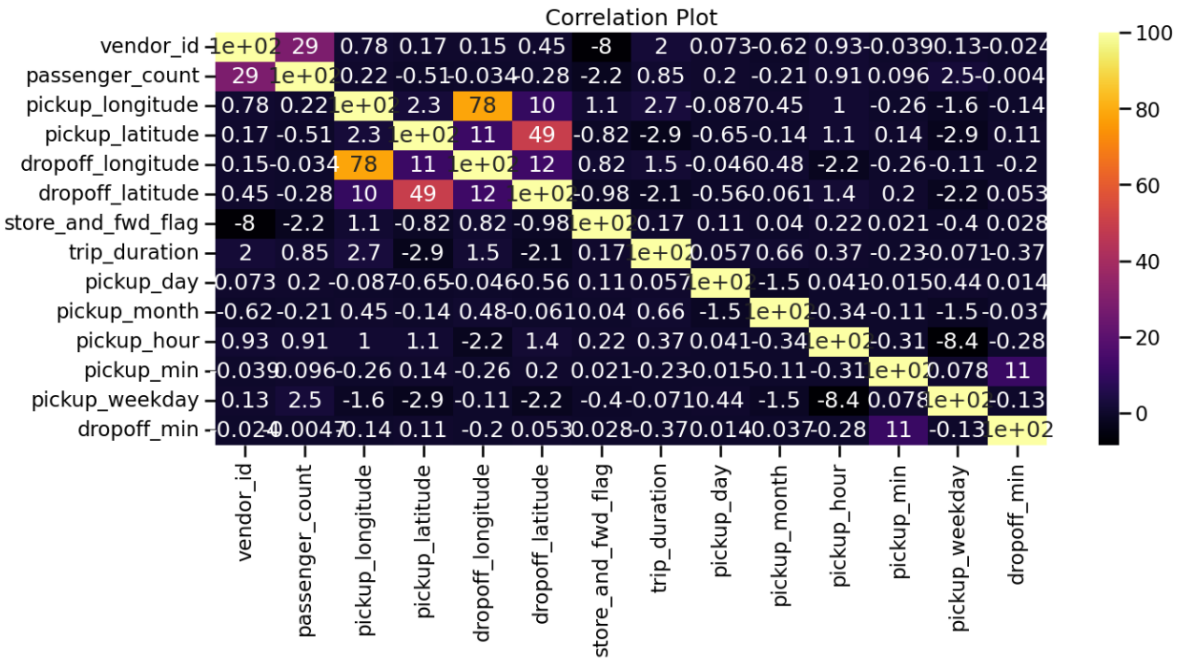
In which hour we get to see maximum pickups? - Rush hours (5 pm to 10 pm), probably office leaving time.



Observations tells us that Fridays and Saturdays are those days in a week when New Yorkers prefer to rome in the city. GREAT !!



2.2.7 Correlation Heatmap:



Chapter 3

Technology Used

3.1. Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

3.2. Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

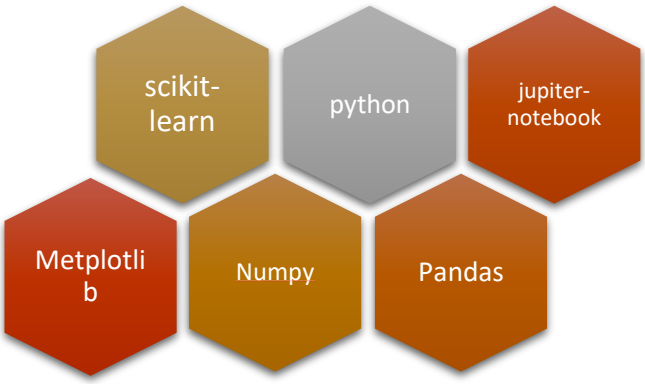


Fig 3. Technology used.

3.3. Pandas

Pandas is a Python library for data analysis. **Started by Wes McKinney in 2008** out of a need for a powerful and flexible quantitative analysis tool, pandas has grown into one of the most popular Python libraries.

3.4. NumPy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. **NumPy was created in 2005 by Travis Oliphant**. It is an open-source project, and you can use it freely.

3.5. Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

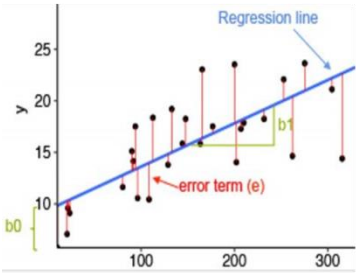
3.6. Scikit-learn.

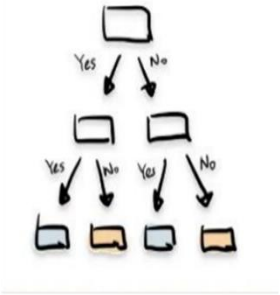
Scikit-learn is a key library for the Python programming language that is typically used in machine learning projects. Scikit-learn is focused on machine learning tools including mathematical, statistical and general purpose algorithms that form the basis for many machine learning technologies. As a free tool, Scikit-learn is tremendously important in many different types of algorithm development for machine learning and related technologies.

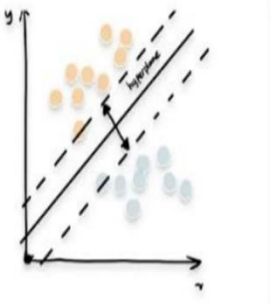
Chapter 4

Machine Learning algorithms:

1. Linear Regression **2. Decision Tree** **3. Support Vector Machine**







4.1 Linear Regression:

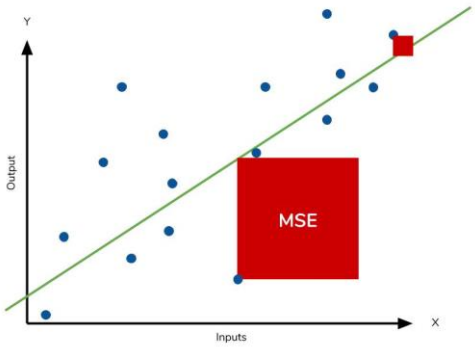
Linear Regression is a regression of dependent variable on independent variable. It is a linear model that assumes a linear relationship between dependent (y) and independent variables (x). The dependent variable
(y) is calculated by linear combination of independent variable (x).

$$Y=B_0+B_1x_1+B_2x_2$$

The cost function for linear regression is given by:

Minimum sum of square error

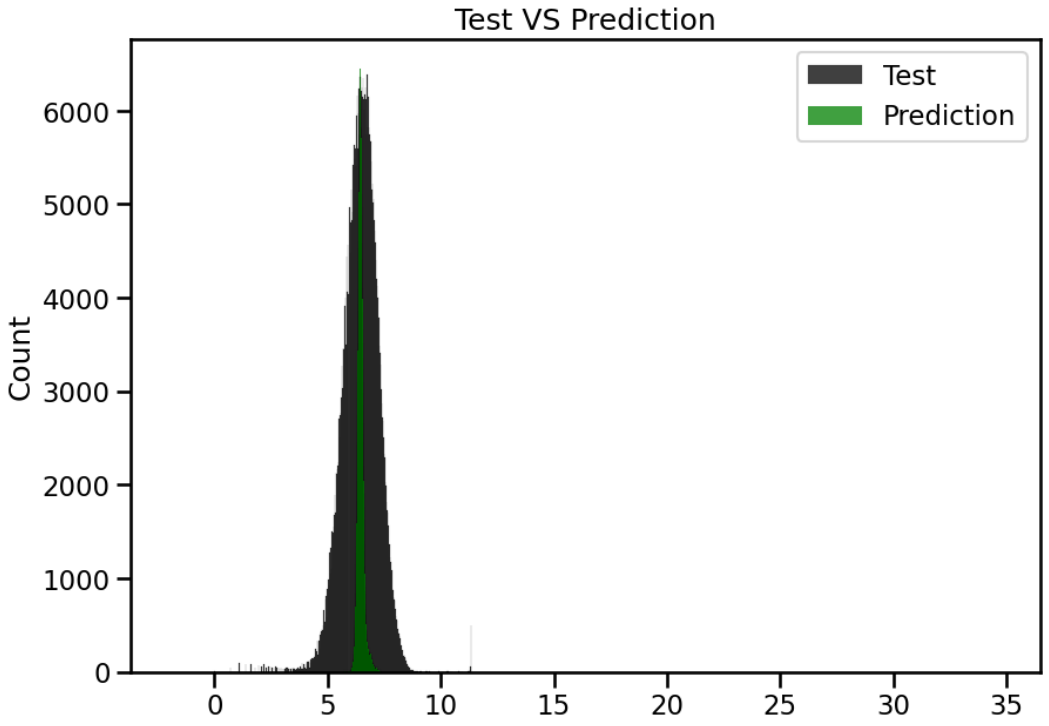
$$MSSE=\sum_1^n (Y_i\, act - Y_i pred)^2$$



```
Training Score :  0.04244450511791209
Validation Score :  0.043892079955855645
Cross Validation Score :  -0.048583442002601875
R2_Score :  -23.10135437604674
```

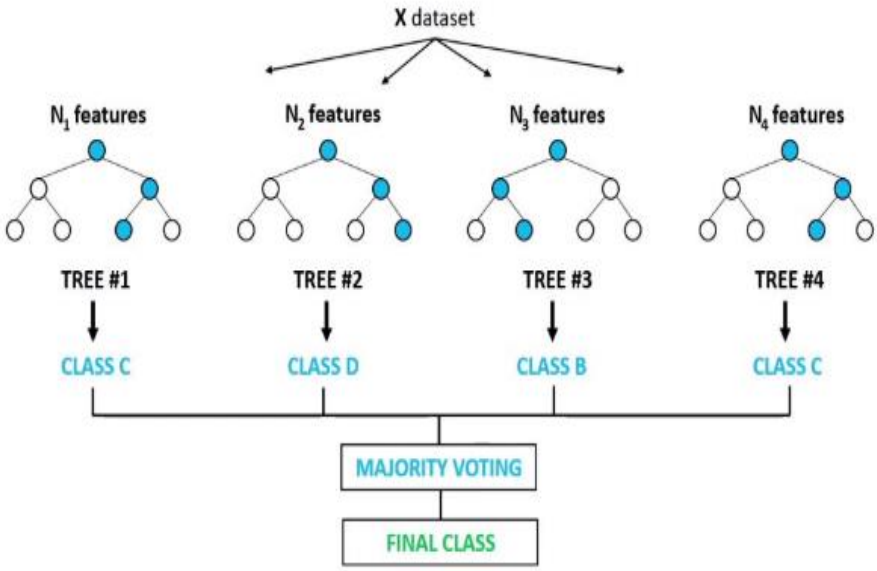
Linear Algorithm: - prediction vs real data

Viz. we can clearly identify that the Linear Regression isn't performing good. The Actual Data (in Grey) and Predicted values (in Yellow) are so much differing. We can conclude that Linear Regression doesn't seem like a right choice for Trip duration prediction.



4.2 Random Forest Regression :

The random forest approach is a bagging method where deep trees, fitted on bootstrap samples, are combined to produce an output with lower variance. However, random forests also use another trick to make the multiple fitted trees a bit less correlated with each other: when growing each tree, instead of only sampling over the observations in the dataset to generate a bootstrap sample, we also sample over features and keep only a random subset of them to build the tree. Sampling over features has indeed the effect that all trees do not look at the exact same information to make their decisions and, so, it reduces the correlation between the different returned outputs. Thus, Random Forest algorithm combines the concepts of bagging and random feature subspace selection to create more robust models.



A Random Forest Algorithm is a supervised machine learning algorithm which is extremely popular and is used for Classification and Regression problems in Machine Learning.

We know that a forest comprises numerous trees, and the more trees more it will be robust.

```
#examining metrics

print ("Training Score : " , est_rf.score(X_train, y_train))

print ("Validation Score : ", est_rf.score(X_test, y_test))

print ("Cross Validation Score : " , cross_val_score(est_rf, X_train, y_train, cv=5).mean())

print ("R2_Score : ", r2_score(rf_pred, y_test))

print ("RMSLE : ", np.sqrt(mean_squared_log_error(rf_pred, y_test)))
```

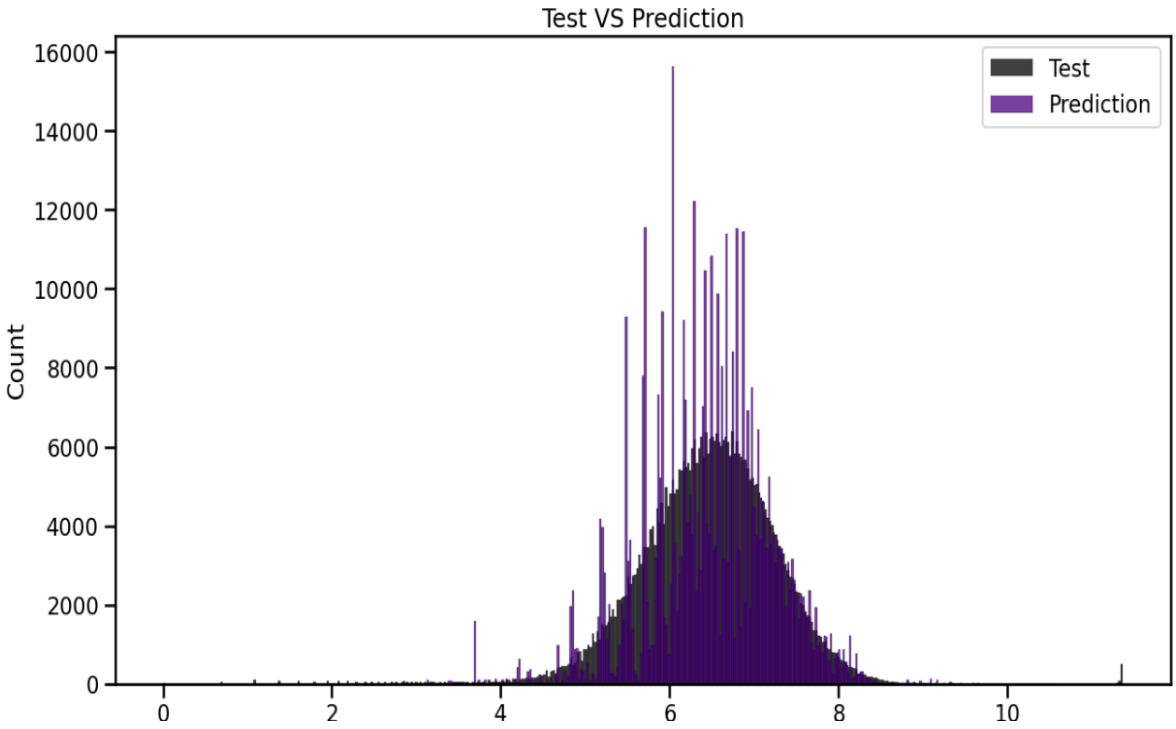
```
Training Score :  0.9305341702955872
Validation Score :  0.9248838628922653
Cross Validation Score :  0.9242455425858781
R2_Score :  0.918176311630572
RMSLE :  0.035755791943398875
```

Random Forest Algorithm :- prediction vs real data:

- From the above Viz. we can clearly identify that the Random Forest Algorithm is also performing good. The Actual Data (in Grey) and Predicted values (in Green)

are as close as possible. We can conclude that Random Forest could be a good choice for Trip duration prediction.

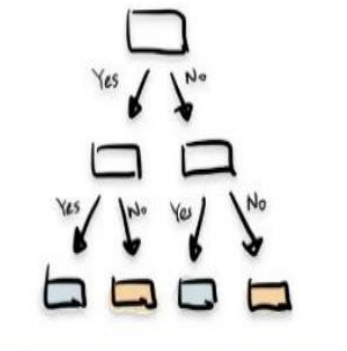
- Similarly, we can Hyper tune Random Forest to get the most out of it.



So, we can see that the random forest algorithm has good accuracy in prediction.

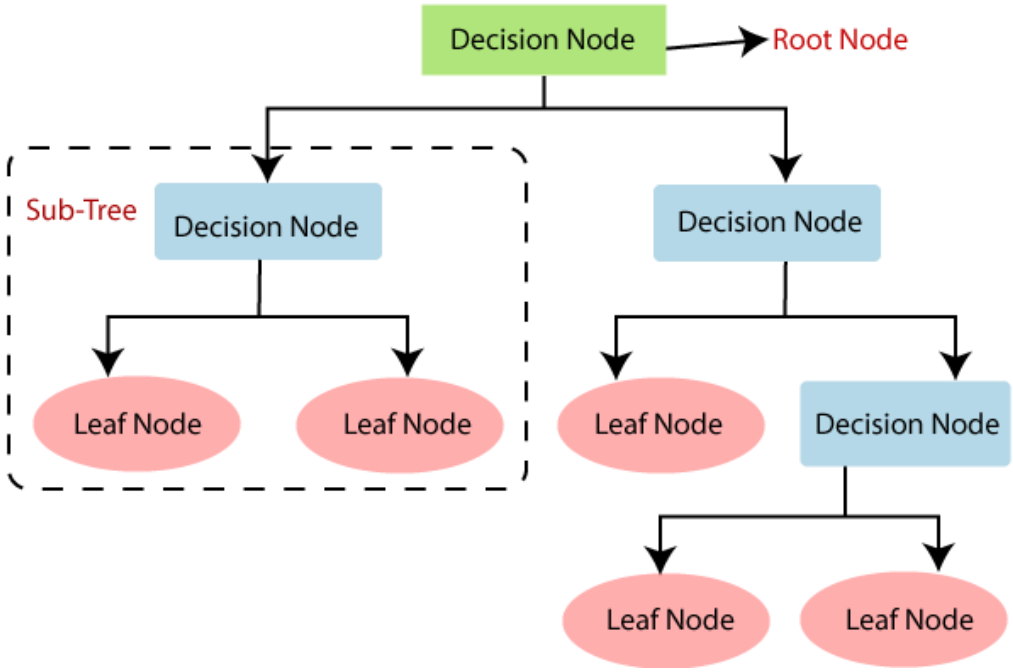
4.3 Decision Tree Regression Algorithm:

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create the model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.



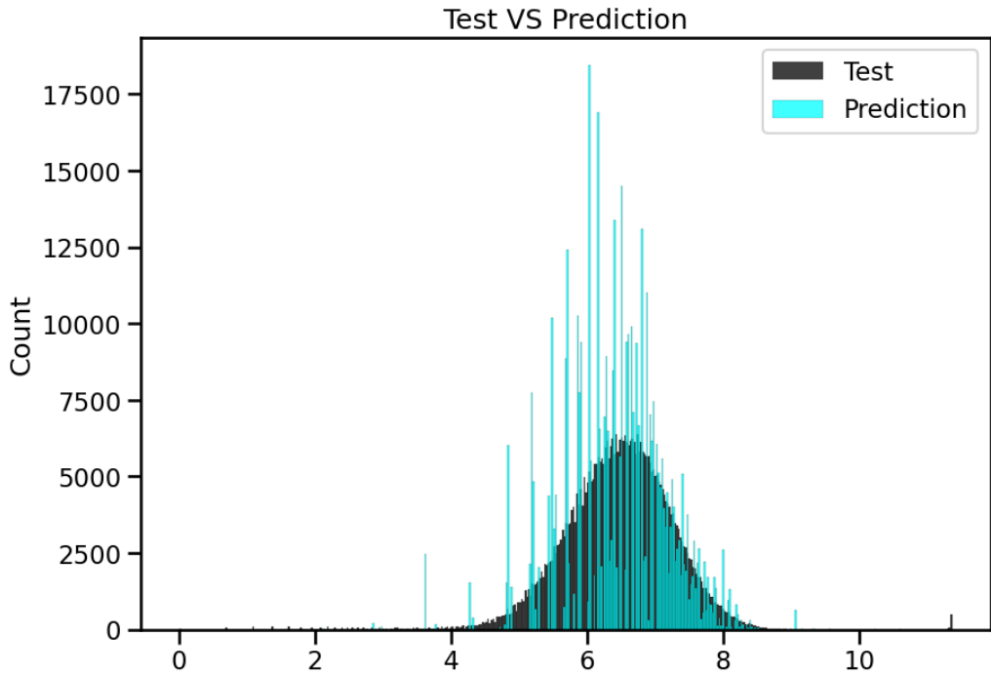
```
#examining metrics

print ("Training Score : " , est_dt.score(X_train, y_train))

print ("Validation Score : ", est_dt.score(X_test, y_test))

print ("Cross Validation Score : " , cross_val_score(est_dt, X_train, y_train, cv=5).mean())

print ("R2_Score : ", r2_score(dt_pred, y_test))

print ("RMSLE : ", np.sqrt(mean_squared_log_error(dt_pred, y_test)))
```

```
Training Score :  0.9258236409034742
Validation Score :  0.9167496153990037
Cross Validation Score :  0.9136198482488055
R2_Score :  0.9102582371818634
RMSLE :  0.03756175968240503
```



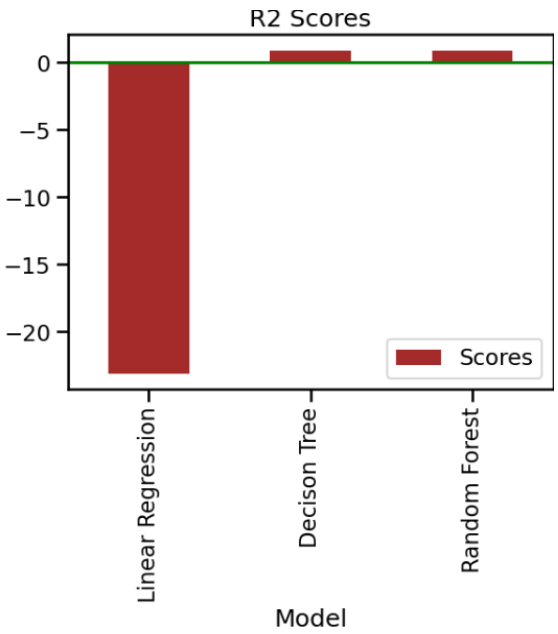
Decision Tree Algorithm:- prediction vs real data:

From the above Viz. we can clearly identify that the Decision Tree Algorithm is performing good. The Actual Data (in Grey) and Predicted values (in Red) are as close as possible. We can conclude that Decision Tree could be a good choice for Trip duration prediction.



4.4 R2 Scores Evaluation:

- R2 Score or R-Squared is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).
- Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

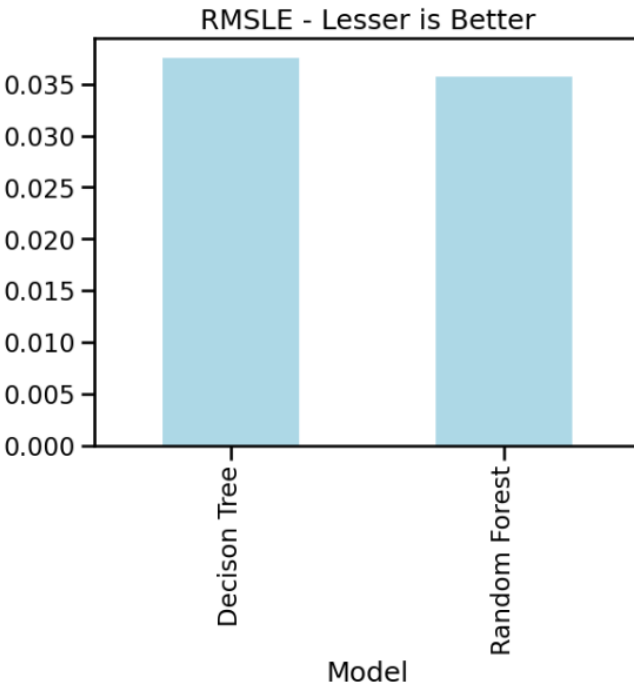


- Although , our Evaluation Metric isn't R2 Score but I'm just plotting them to check the Good Fit.
- We're getting good fit score for Decision Tree and Random Forest , i.e, close to 1.0.

4.5 RMSLE Evaluation:

- RMSLE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.
- With RMSLE we explicitly know how much our predictions deviate.
- Lower values of RMSLE indicate better fit with lesser LOSS

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$$

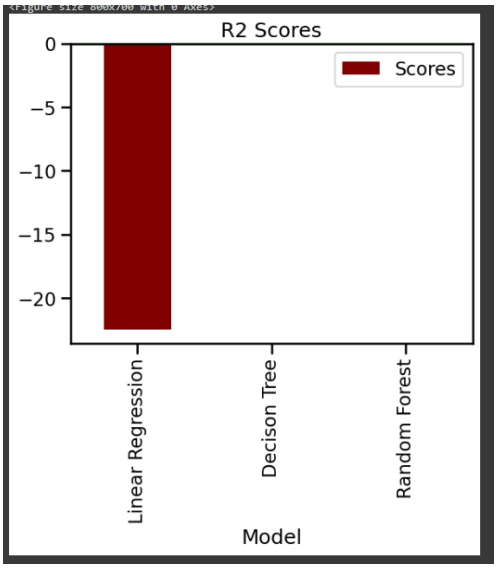


- Remember our NULL RMSLE : 0.1146 as a benchmark to beat.

- We can observe from above Viz. that our Decision Tree model and Random Forest model are good performers. As, Random Forest is providing us reduced RMSLE, we can say that it's a model to Opt for.

4.6 Second Approach - Without PCA (R2 Scores Evaluation):

- Another approach we could go with is without PCA, just Standard Scaling Dataset and applying our Algorithms.
- The approach can give us better idea of what works better for us.
- This approach might take great amount of computational resources and time, it will be good if we can run this on Google's Collaboratory, that will eliminate huge computational stress on our system as the program will be running on Cloud.




```
Training Score :  0.043473061426662074
Validation Score :  0.045092484959132983
Cross Validation Score :  -0.048701119150320826
R2_Score :  -22.45617356992333
RMSLE :  0.11228235950685247
```

```
Training Score :  0.4643053926882248
Validation Score :  0.4579574624226187
Cross Validation Score :  0.45745671106348684
R2_Score :  -0.16168520580612133
RMSLE :  0.08783882677794577
```

```
Training Score :  0.4770512476779144
Validation Score :  0.47142456987537174
Cross Validation Score :  0.47093026897207546
R2_Score :  -0.17083406013290636
RMSLE :  0.08692710581035555
```

We can see that the SVM classifier is giving the best accuracy.

Chapter 5

5.1 Challenges Faced:

- Reading the dataset and understanding the meaning of some columns.
- For answering some of the questions we had to understand
- NYC Taxi time prediction model that how they work.
- Designing multiple visualizations to summarize the information in the dataset and successfully communicate the results and trends to the reader.

4.2 Future work:

As a part of the future work, the Multi-layer Perceptron model could be auto tuned to further learn and determine which features need to get joined to detect numerous interactions between them as needed. Moreover, variabilities and quantities related to the various location features might also be computed in the upcoming research in order to localize the traffic-based effects on the taxi prediction coordinates. Speed limitations-based features could later be incorporated alongside to comprehend better analysis of the datasets. At last, enhancements to the K-Means Clustering algorithm could be provided by encompassing additional features such as distance to the closest metro station, number of bars and eateries in a given zone, etc. so as to exploit comparative qualities belonging to various zones. This would also ensure the rightful evaluation of various clusters in which each data point falls such that it fills in as an extra vital element for our models.

5.3 Conclusion:

- Observed which taxi service provider is most Frequently used by New Yorkers.
- Found out few trips which were of duration 528 Hours to 972 Hours, possibly Outliers.
- With the help of Tableau, we're able to make good use of Geographical Data provided in the Dataset to figure prominent Locations of Taxi's pickup / dropoff points.
- In this project, we tried to predict the trip duration of a taxi in NYC.
- We are mostly concerned with the information of pick-up latitude and longitude and drop off latitude and longitude, to get the distance of the trip.
- Hyperparameter tuning doesn't improve much accuracy.
- Also, found out some Trips of which pickup / dropoff point ended up somewhere in North Atlantic Sea.

- ❖ Passenger count Analysis showed us that there were few trips with Zero Passengers.
- ❖ Monthly trip analysis gives us a insight of Month – March and April marking the highest number of Trips while January marking lowest, possibly due to Snowfall

4.4 References:

- Blog.Jovian.ai
- Analytics Vindhya
- Kaggle.com